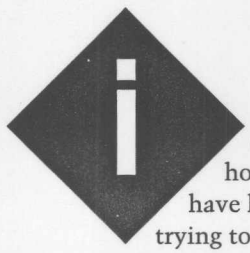
Analyst 2 Data Line Monitor

FEATURE ARTICLE

Bill Payne

You make your serial connections, run the appropriate software on both ends, and nothing happens. What do you do next? If you have an Analyst 2 data line monitor, you'll have your connection buzzing in no time.



I cannot recount how many times I have had trouble trying to troubleshoot a PC's serial data link. Whether the external device is a modem, a printer, a serial-to-parallel converter, or another computer, the problem is that I have needed to see the actual data flowing between the two devices. I have tried to use a breakout box with LED indicators, but it only works for slowly changing control-signal leads. If the problem is caused by an incorrect transmission rate, fast glitches occurring on the control-signal leads, or a mismatched character set, simple tools will not do the job. Such problems force me to pull out my Analyst 2 Data Line Monitor (Analyst 2) from the desk drawer.

In the field of data communications, there are protocol analyzers and there are data line monitors. Protocol analyzers are very sophisticated devices which enable a user to monitor the transmission on a serial communications link and to break down that transmission into the specific parts of the software protocol. A data line monitor passively bridges an RS-232 communications link, capturing and then displaying the serial data stream in a humanly intelligible form.

Analyst 2 can be used to monitor all asynchronous, synchronous, and user-defined, bit-oriented communications systems at speeds up to 38,400 bps. During the capture of data, it can switch the display of the serial data stream from a character-oriented format to a hexadecimal format by pressing a single key. I can set up Analyst 2 to look for specific characters before it starts or stops capturing the serial data stream. I can set it up to monitor specific control-signal leads for a transition to control capturing the stream or to stop it when a transmission error occurs.

Analyst 2 has the ability to perform signal distortion analysis, which checks the communications line for the relationship between the actual and theoretical signal durations. Signal distortion can occur in any communications system if the reference clock for the transmitter or receiver becomes skewed in any way due to age, improper frequency selection, or any of a dozen other factors. Analyst 2 can measure all types of signal distortion on a communications line.

Analyst 2 has the ability to measure the time delay between various hardware-handshake signal leads on the DB-25 interface. In this mode, Analyst 2 will actively stimulate the request-to-send (RTS) signal lead and wait for a corresponding response on the selected signal lead. clear-to-send (CTS) delay times for turning on and off can be measured relative to the assertion/deassertion of the RTS signal lead. Also, the time between the deassertion of the RTS signal lead until the detection of the carrier detect (CD) signal can be measured. This time duration is of paramount importance in half-duplex communications systems where it is commonly referred to as *turn around time.*

Analyst 2 also has the ability to simulate raw data traffic in all communications modes. This function can be used to validate the integrity of a communications line under test by inserting a known message into it. It can be used to validate the performance of printers, modems, and even software communications programs. Analyst 2 accomplishes this by outputting the Quick Brown Fox (QBF) message. The QBF message contains all the letters of the alphabet, numbers 0–9, a carriage return, and a line feed. The message can be transmitted in the ASCII, EBCDIC, or Baudot character sets. Analyst 2 pauses approximately 250 ms at the

end of each transmission before repeating the pattern.

The most powerful feature of Analyst 2 is the ability to review the captured serial data stream at the user's convenience. This is accomplished through the use of battery-backed memory for storing the captured data. Using the Review mode, I can change the display parameters at any time without affecting the capture memory contents. For example, I may have originally specified that the serial data being captured was transmitted least-significant bit first. Through Review mode, I can change the displayed bit order to most-significant bit first without affecting the original serial data in the nonvolatile capture memory. While reviewing the captured data, I can switch between the selected and hexadecimal character set with a single keystroke.

HARDWARE DESCRIPTION

The complete schematic for the Analyst 2 is shown in Figure 1.

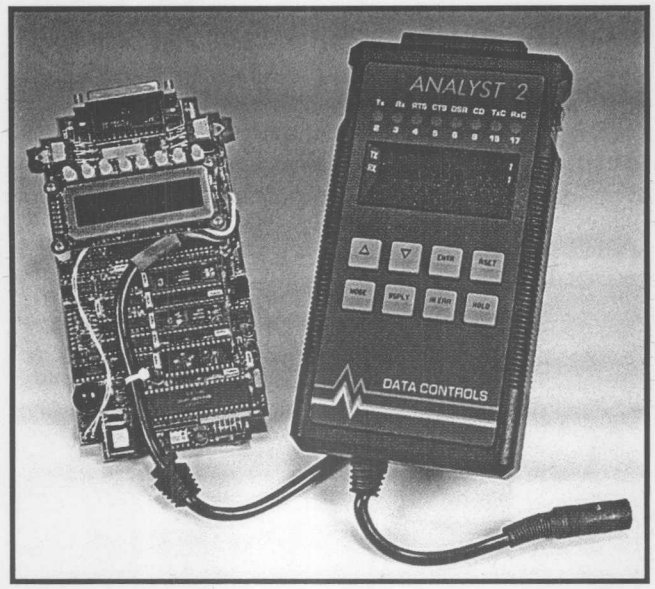
From the outside, you see the Analyst 2's user interface, which is made up of a 2-line, 32-character backlit LCD, 8 tricolor LEDs, and an 8-button keypad. The 8 LEDs represent the various EIA RS-232 interface leads in real-time for data terminal equipment (DTE). The following signal leads are monitored and displayed:

Pin 2: Transmit Data (TxD)
Pin 3: Receive Data (RxD)
Pin 4: Request To Send (RTS)
Pin 5: Clear To Send (CTS)
Pin 6: Data Set Ready (DSR)
Pin 8: Carrier Detect (CD)
Pin 15: Transmit Clock (TxC)
Pin 17: Receive Clock (RxC)

During the power-on self-test (POST), the TxD, RxD, TxC, and RxC LEDs glow red indicating a mark (idle) condition. The RTS, CTS, DSR, and CD LEDs glow green indicating a space (off) condition. If an error occurs in the POST, the LEDs will flash, the audible alarm will sound, and the LCD will display an error code.

The 8-button keypad is arranged as two rows of four keys each. The first row is composed of the Up Arrow, Down Arrow, Enter, and Reset keys. The second row is composed of the Mode, Display, Insert Error, and Hold keys. These keys function as follows:

- Down Arrow—in the setup phase of the operating mode, this key scrolls the menu selections in reverse order. In the Review mode, it reverses the order of the captured data to be written to the LCD.
- Up Arrow—in the setup phase of an operating mode, this key scrolls the menu selections forward. In the Review mode, it displays the captured data to be written to the LCD in the order it was captured.
- Enter—this key is used exclusively for entering the information displayed on the LCD into the operating mode setup configuration.
- Reset—during the setup phase of an operating mode, this key brings back the Main Menu. During an actively running test, Reset restarts the test.
- Mode—this key is active during the setup and operation of a test and causes an immediate return to the Main Menu when pressed.
- Display—only available during Monitor mode, this key toggles the display from current to hexadecimal character set.
- In Err—this key is not functional.
- Hold—this key stops the capture of serial data and transfers the user to Review mode for editing of the captured data.



Internally, Analyst 2 is composed of a Zilog Z86C81 (Z8) microcomputer, Zilog Z8030 serial communications controller (ZSCC), up to 64 KB of SRAM, 32 KB of program memory, an LM556 dual timer, a Texas Instruments TL7705A reset generator, and various RS-232 drivers and receivers.

Because of its versatility, I chose to use the Zilog Z8 as the processor in Analyst 2. The Z8 has a multiplexed address data bus which can be directly attached to any of the Zilog ZBUS peripherals. It supports up to 64 KB of external program memory, up to 64 KB of external data memory, and has 128 internal registers.

The internal register space is divided into 124 general-purpose registers, 16 CPU and peripheral control registers, and 4 I/O port registers. Each register is 8 bits wide and can be used as accumulators, address pointers, indexes, data, or stack registers.

A register pointer logically divides the register file into 9 groups of 16 working registers which enables the program operating in one register bank to context switch to another register bank in the event of an interrupt condition. It also provides a simple multitasking operating system to be developed which uses a separate register bank for each operating task. When a task switch is done, only the register pointer is updated, and no pushing or popping of the stack contents is needed.

Port 1 on the Z8 microcomputer is configured as the multiplexed address/data bus. This port is wired directly to the Z80C30 SCC and to a 74ALS373 for latching the lower 8 bits of the address bus. Port 0 is configured as

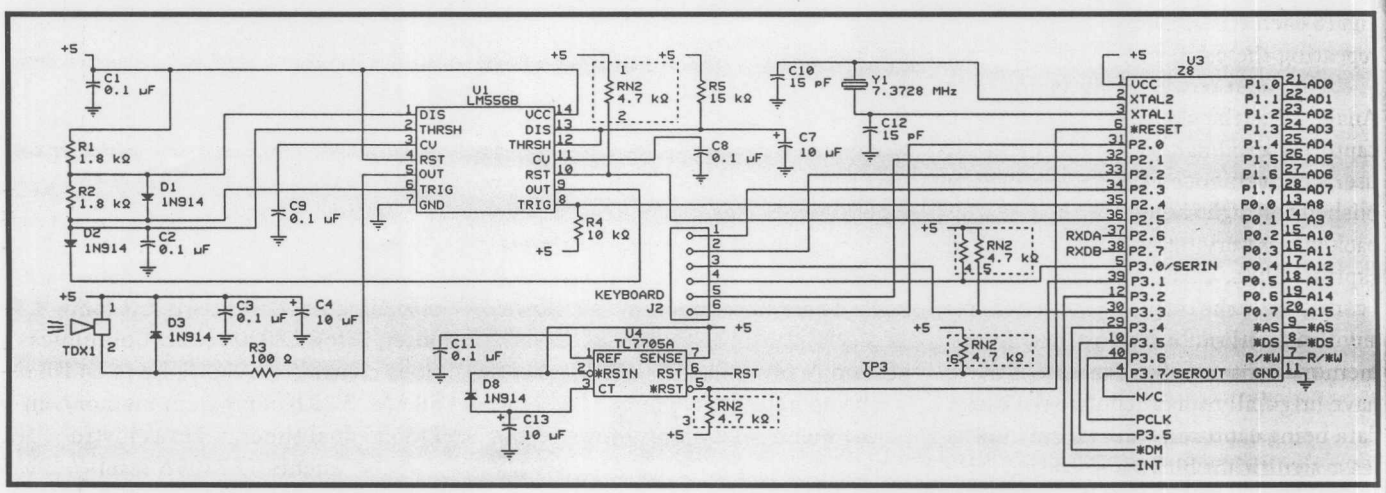


Figure 1a—*At the core of the Analyst 2 is a Zilog Z8 processor. The TL7705A generates a clean reset for the unit while an LM556 dual timer handles the beeper. The keyboard goes right into the processor to be handled by the firmware. Finally, note that the processor generates the UART's 3.6864-MHz clock on its PCLK output.*

address lines A8–A15. All pins on this port are individually pulled up to VCC through a 10 kΩ resistor, a necessary step since these pins are undefined during reset. Port 2 is bit definable and thus is used for multiple purposes. Pins 0–3 are used as the active scan lines for the keyboard. Pins 4–5 are used to control the 556 timer for the piezo alarm. Pins 6–7 are used to monitor the serial interface during the distortion tests.

The Z80C30 ZSCC is a dual-channel, multiprotocol data communications peripheral. The device contains on-chip baud rate generators and digital phase-locked loops for each channel. The ZSCC is designed to handle all asynchronous, byte synchronous, and bit synchronous communications protocols. In addition, the device is capable of generating and checking Cyclic Redundancy Check (CRC) codes in all synchronous modes and provides complete error detection in all asynchronous modes.

The ZSCC is connected to the multiplexed address/data bus from the Z8 microcomputer which enables it to be accessed as a ZBUS peripheral at a much higher rate than the nonmultiplexed version of this part (Z85C30). The Z8 provides the ZSCC with a 3.6864-MHz clock which is derived from the Z8 7.3728-MHz crystal clock. This clock frequency allows the ZSCC

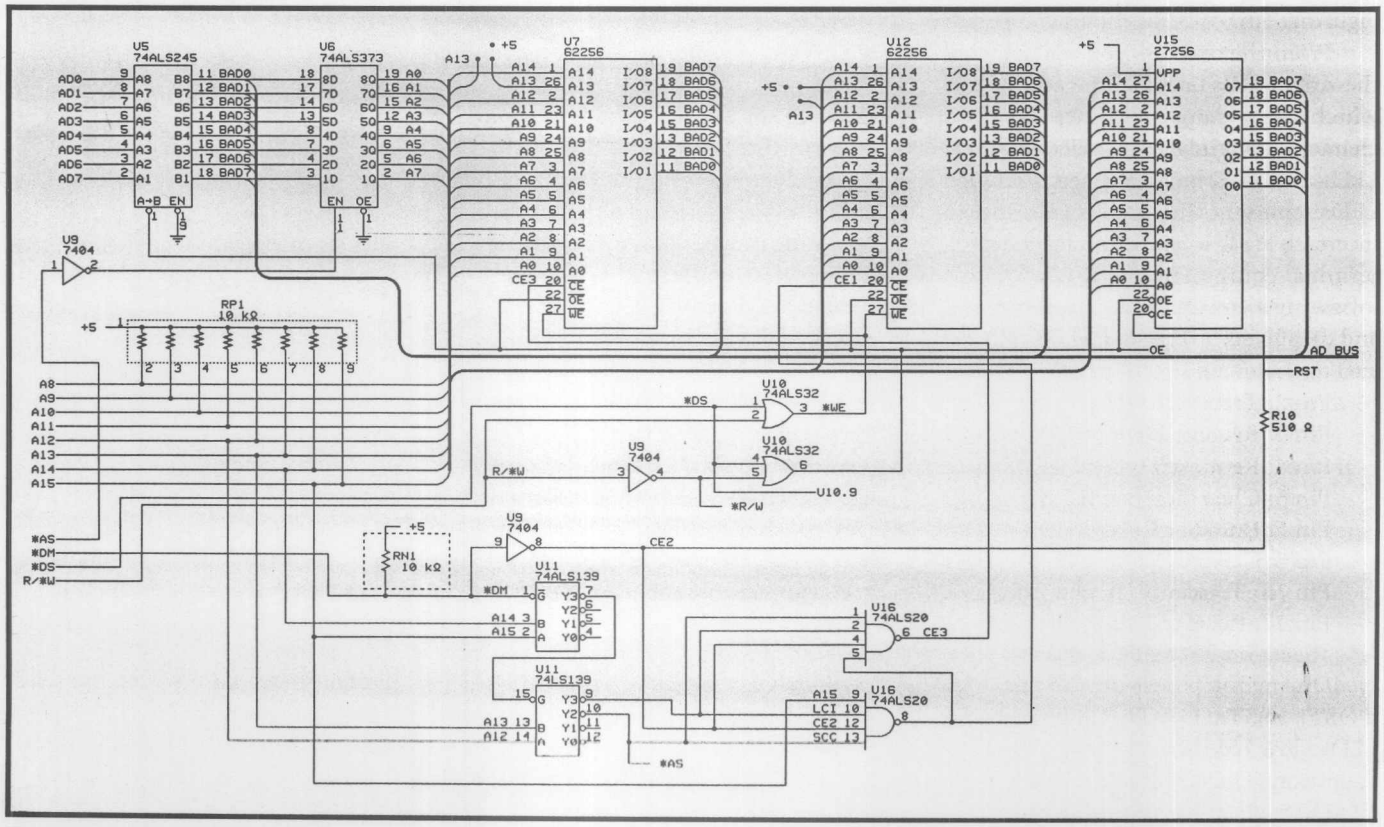


Figure 1b—*The Analyst 2 contains 32 KB of EPROM and 64 KB of RAM. All memory and peripheral decoding is done using discrete logic rather than programmable devices.*

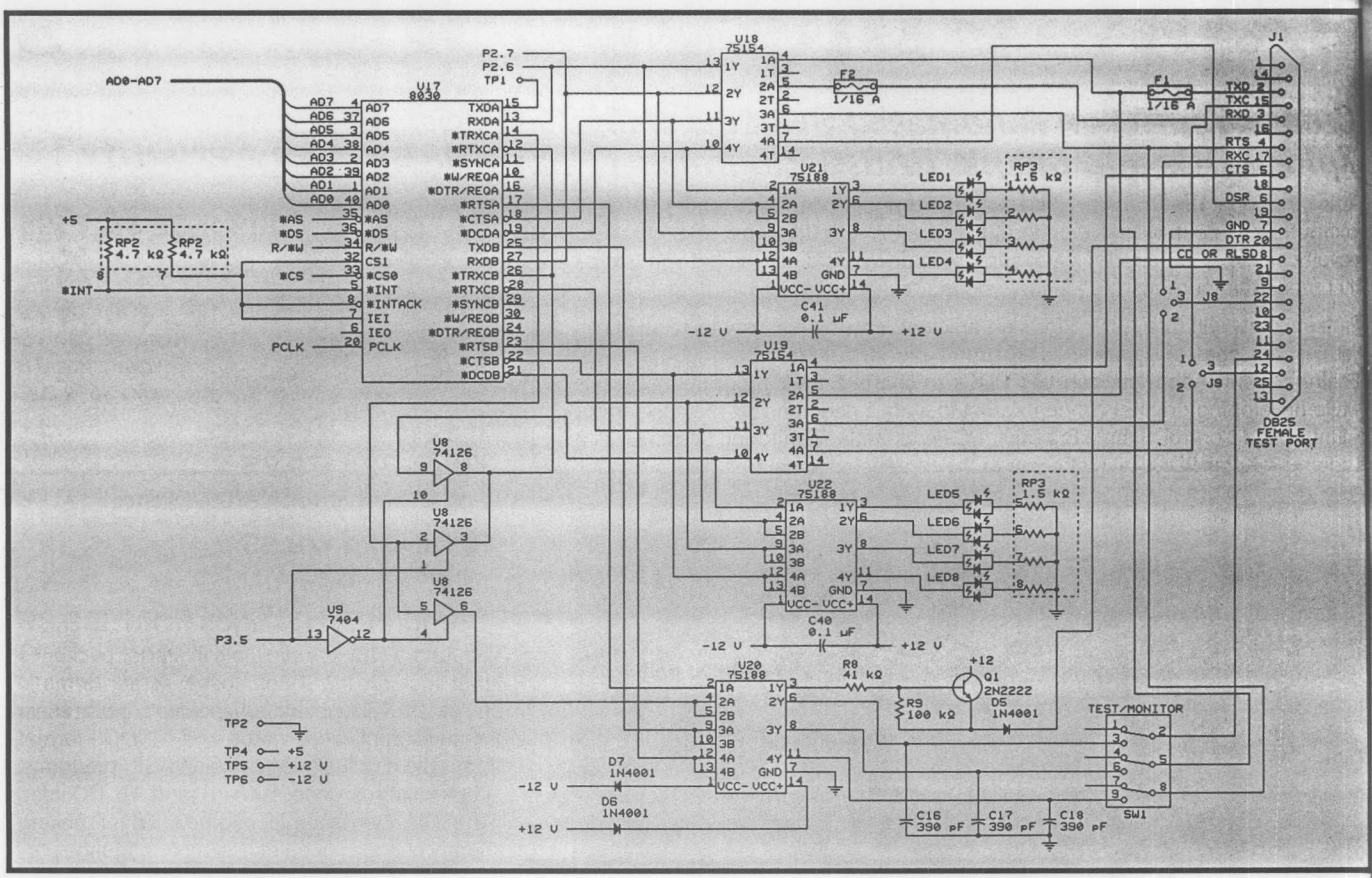
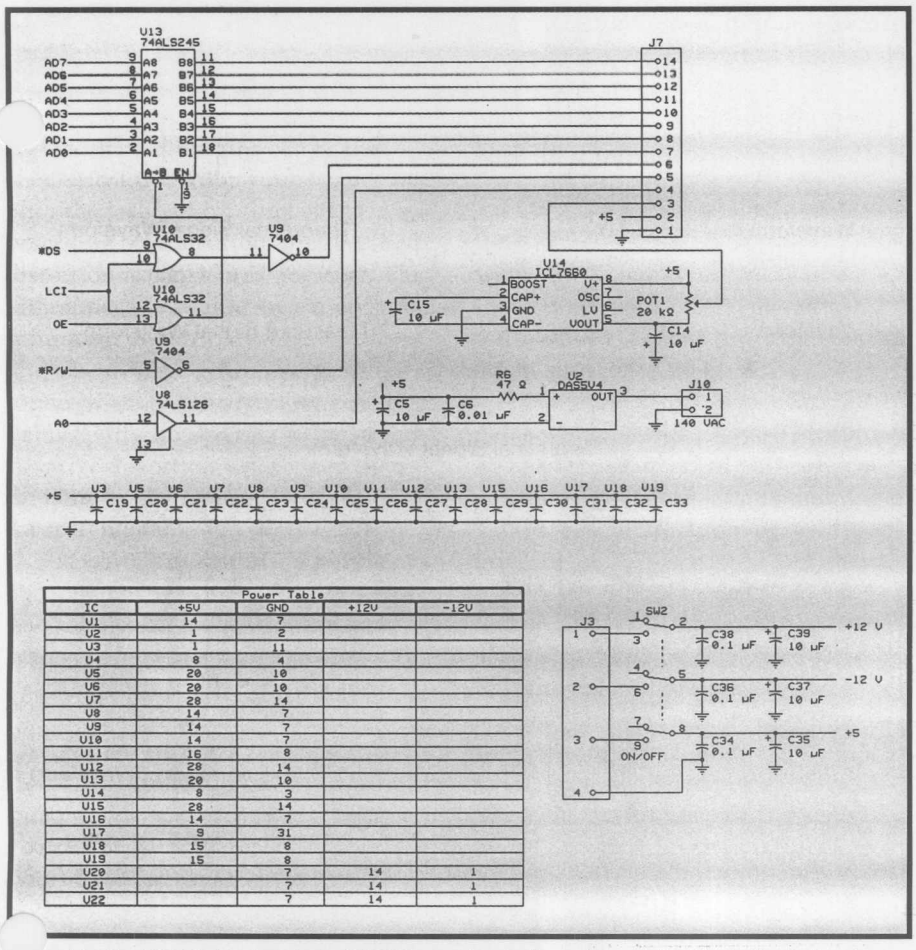


Figure 1c—*The Z80C30 serial communications controller is designed to handle all asynchronous, byte synchronous, and bit synchronous communications protocols.*



Power Table

IC	+5V	GND	+12V	-12V
U1	14	7		
U2	1	2		
U3	1	11		
U4	8	4		
U5	20	10		
U6	20	10		
U7	28	14		
U8	14	7		
U9	14	7		
U10	14	7		
U11	16	8		
U12	28	14		
U13	20	10		
U14	8	3		
U15	28	14		
U16	14	7		
U17	9	31		
U18	15	8		
U19	15	8		
U20		7	14	1
U21		7	14	1
U22		7	14	1

Figure 1d—*The LCD interface includes an ICL7660 power inverter to create a negative bias for the display plus a DAS5V4 DC-to-AC converter to drive the EL backlight.*

to meet all timing requirements for data speeds ranging from 50 through 38,400 bps.

During a monitoring operation, both receive channels of the Z80C30 are used. Channel A is attached to Pin 2 and Channel B is attached to Pin 3 on the DB-25 interface connector. This configuration enables the unit to monitor both the transmit and receive sides of a serial communications session. When working with a protocol which requires the use of a phase-locked loop for clocking information recovery, Channel A clocking is connected through a 74HC125 to the clock-output pin from Channel B. In this configuration, the baud rate generator for Channel B is used to provide the 32× clock for the phase-locked loop in Channel A.

The NE556 timer is a dual 555 timer in a single DIP package. One 555 timer is used as the frequency generator to drive the piezo transducer. The other 555 timer is used as a period controller for the frequency generator. With this design, the Z8 microcomputer only needs to toggle a port pin to reset or fire the piezo alarm. The period of the alarm is fixed by the second 555 timer to approximately 0.5 s. This alarm is used as an audible feedback for keypad entries and for signaling.

The TL7705A is a voltage-controlled reset device which will hold the Z8 microcomputer in a reset state whenever the power to the device is out of tolerance. Once the power supply has stabilized, the TL7705A continues to assert the Reset line to the processor for approximately 250 ms. This allows the Z8 enough time for the oscillator to stabilize before fetching the first instruction from EPROM.

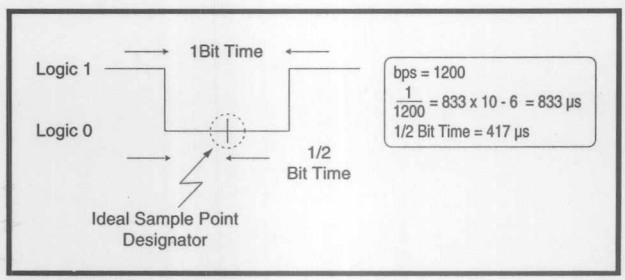


Figure 2—*At 1200 bps, a single bit time lasts 833 µs while half a bit time is 417 µs.*

SOFTWARE DESCRIPTION

Analyst 2's various modes of operation are Data Monitor, Review, Distortion, Time Delay, and Simulate.

•Monitor

After all setup configurations have been entered into the Analyst 2 for the Monitor mode, the user will be prompted with one final menu selection:

Execute—This selection starts data capture as specified by the user. Serial data displays on the LCD as it starts flowing into the capture memory. At higher baud settings, capture memory fills faster than display speed. When capture memory is full, the alarm sounds three times signifying that it is ready to begin capture again.

Dsply—During the capture process, the Dsply (display) key toggles between the selected and hexadecimal character set without affecting the data capturing in progress.

Hold—This key immediately stops incrementing the display, and all capturing functions cease. It terminates a capture prior to full capture memory. Rset (reset) invokes the Review mode, allowing the user to scroll forward and backward through the captured serial data stream.

•Review

The Review mode displays the captured serial data in the manner specified by the menu selections. If an error condition, such as a parity error or framing error, was detected in the Monitor mode, it displays on the LCD when in the Review mode. The Up and Down Arrow keys are used to control the direction of the displayed data. If either key is held for four character cycles, Analyst 2 goes into an automatic state

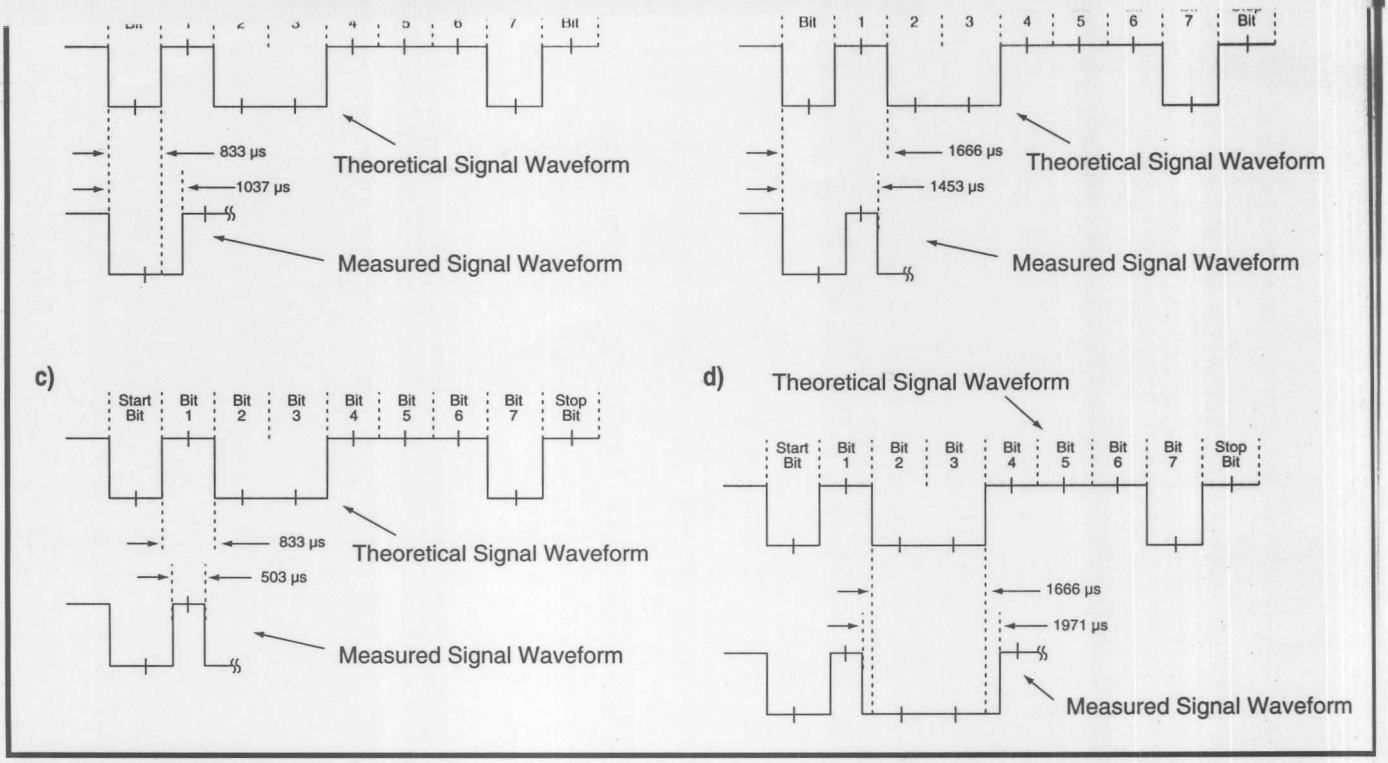


Figure 3—*Gross distortion measures the time between when a transition occurs and when it should occur. Two examples include positive distortion (a) and negative distortion (b). Isochronous distortion represents the difference between the measured and theoretical width of a bit time. Two examples include negative distortion (c) and positive distortion (d).*

for reviewing the captured serial data. At any time, the Dsply button can change the display to a hexadecimal format. Analyst 2 beeps once when it reaches the end of the nonvolatile capture memory.

The display immediately stops if the Hold key is pressed during the review process. Pressing the Rset key resets the display to the beginning of the capture memory and restarts the operation.

•Distortion

Distortion is defined as a deformity of the data communications signal compared to its theoretical timing parameters. Gross, isochronous, and bias distortion may be present on a data communications line, each relating a specific set of measurements on the signal waveform to the theoretical timing parameters.

Figure 2 defines the terminology. It is assumed that the measurements are taken on a data communications line operating at 1200 bps.

The 1200-bps speed implies a bit time of 883 µs. This time applies to all bits in the data stream including the start, parity, and stop bits in an asynchronous communications environment. Most communications systems determine the state of the bit by sampling at the middle of the bit time. For my example of 1200 bps, a midsample occurs approximately 417 µs after the falling edge of the bit-time period.

Gross distortion measures the time between when a transition occurs and when it should occur (see Figure 3a). The reference point is an arbitrarily chosen transition from logic 1 to logic 0. The transition time is measured from the reference point to a transition from logic 0 to logic 1. The measured time is compared to the theoretical time and a percentage difference is calculated and displayed on the LCD:

$$\frac{MBT - TBT}{TBT} \times 100 = \%\,Gross\,Distortion$$

where *MBT* represents the Measured Bit Time and *TBT*, the Theoretical Bit Time.

For example, if the transition edge that ends the Start Bit and starts the Bit 1 time period should occur at the theoretical time of 833 µs after the falling edge of the Start Bit period, and the measured time period is actually 1037 µs after the falling edge of the Start Bit, then the gross distortion is 24%:

$$\frac{1037\,\mu s - 833\,\mu s}{833\,\mu s} \times 100 = 24.4\%$$

When the measured transition time of the specified bit begins after the theoretical transition time, but before the half-bit time of the next consecutive bit transition, the specified value is considered positive. The largest positive value measured is the maximum gross distortion result.

As another example in Figure 3b, the transition edge that ends the Bit 1 period and starts the Bit 2 period should occur at the theoretical time of 1666 µs after the falling edge of the Start Bit. The measured time period for the communications line being tested is 1453 µs after the falling edge of the Start Bit, and the gross distortion is:

$$\frac{1453\,\mu s - 1666\,\mu s}{1666\,\mu s} \times 100 = -12.8\%$$

When the measured transition time of the specified bit begins before the theoretical transition time begins, but after the last half-bit time of the preceding bit transition, the specified distortion is considered negative. The largest negative value measured is the minimum distortion value.

The average gross distortion is found by adding the positive distortion values with the absolute value of the negative distortion values, and then dividing by the total number of measurements.

Isochronous distortion represents the difference between the measured and theoretical pulse width of a bit-time. Logic 1s and 0s are treated the same, and the levels between the transitions are ignored. Measurement samples are arbitrarily started at the first logic 0 transition in the communications data stream. The transition time is measured from this reference point until another transition is detected. The measured time is then compared to the theoretical time, and the percentage difference is calculated and displayed on the LCD using the formula:

$$\frac{MPW - TPW}{TPW} \times 100 = \%Distance$$

where *MPW* represents Measured Pulse Width and *TPW*, Theoretical Pulse Width. Ten bit times of live data are measured for this test.

In Figure 3c, the first measured pulse width after the logic 0 bit time is 503 µs. The theoretical pulse width is calculated as 833 µs (again assuming 1200 bps). The isochronous distortion is:

$$\frac{503\,\mu s - 833\,\mu s}{833\,\mu s} \times 100 = -39.6\%$$

The negative value calculated shows that the pulse width measured for the selected bit rate is 39.6% less than the theoretical pulse width.

Figure 3d offers another illustration. The next measured pulse width after the logic 1 bit time is 1971 µs. The theoretical pulse width is calculated as two times the 833-µs bit time or 1666 µs, and the isochronous distortion is:

$$\frac{1971\,\mu s - 1666\,\mu s}{1666\,\mu s} \times 100 = 18.3\%$$

The positive value calculated shows that the pulse width measured for the selected bit rate is 18.3% larger than the theoretical pulse width.

Bias distortion represents the difference in the time duration of logic 1s (marks) and logic 0s (spaces) in the data stream, and is calculated by:

$$\frac{MPW - SPW}{MPW + SPW} \times 100 = \%Bias$$

where *MPW* represents Mark Pulse Width and *SPW*, the Space Pulse Width. This test is used in asynchronous systems to validate the performance of a transmitting device. Although the alternating mark-space pattern is the preferred pattern to use on a test system, it also works with normal data.

In Figure 4a, the measured pulse width for the logic 0 bit time is 1375

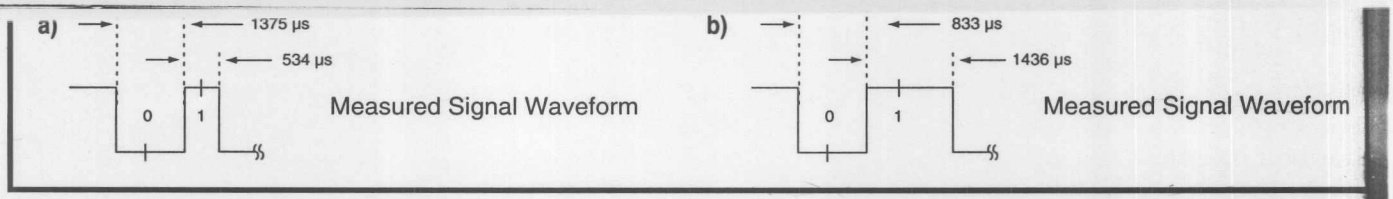


Figure 4—*Bias distortion represents the difference in time duration of logic 1s and logic 0s in the data stream. Examples include negative distortion (a) and positive distortion (b).*

µs. The measured pulse width for the logic 1 bit time is 534 µs. Since bias distortion is concerned only with the ratio of a mark to a space, no theoretical bit times are necessary (Analyst 2 still requires the user to enter the speed of the communications line, but only to properly setup the interface). In this example, the bias distortion is:

$$\frac{534\,\mu s - 1375\,\mu s}{534\,\mu s + 1375\,\mu s} \times 100 = -44.1\%$$

The negative bias distortion shows that the ratio of the pulse widths measured for the selected bit rate are 44.1% larger for logic 0 than the pulse width for the logic 1.

In Figure 4b, the measured pulse width for logic 0 bit time is 833 µs, and the measured width for logic 1 bit time is 1436 µs. Its bias distortion is:

$$\frac{1436\,\mu s - 833\,\mu s}{1436\,\mu s + 833\,\mu s} \times 100 = 26.6\%$$

The positive value calculated shows that the ratio of the pulse widths measured for the selected bit rate are 26.6% smaller for the logic 0 pulse width than for the logic 1 pulse width.

•Time Delay

The Time Delay test is used to measure the internal delay on various control signal leads within the RS-232 interface. Analyst 2 actively stimulates a control signal and waits for a response. This sequence is repeated for 10 samples before the test completes. This function is useful for testing a modem's internal delays between signals such as RTS and CTS. Analyst 2 can measure time intervals including RTS on to CTS on, RTS off to CTS off, and RTS off to CD on.

If the selection is RTS on to CTS on, the unit asserts the RTS signal and waits for the CTS signal to go to an active state (see Figure 5a). This process is repeated for ten sample periods.

If the selection is RTS off to CTS off, the unit deasserts the RTS signal and waits for the CTS signal to go to the inactive state (see Figure 5b).

If the selection is RTS off to CD on, the unit deasserts the RTS signal and waits for the CD signal to go to the active state (see Figure 5c).

•Simulate

The Simulate mode is used to generate known data traffic on a communications line, which is useful in validating the integrity of printers, modems, or even terminal emulation software. Analyst 2 outputs the Quick Brown Fox message for the test.

DATA COMMUNICATIONS FORMATS

The framing selections supported by Analyst 2 are asynchronous, synchronous with one synchronization character, synchronous with two synchronization characters, and synchronous data link control (SDLC/HDLC).

Asynchronous communication uses what's known as *character-framed data* in which each transmitted character has a start bit, 7 or 8 data bits, and 1 or more stop bits (see Figure

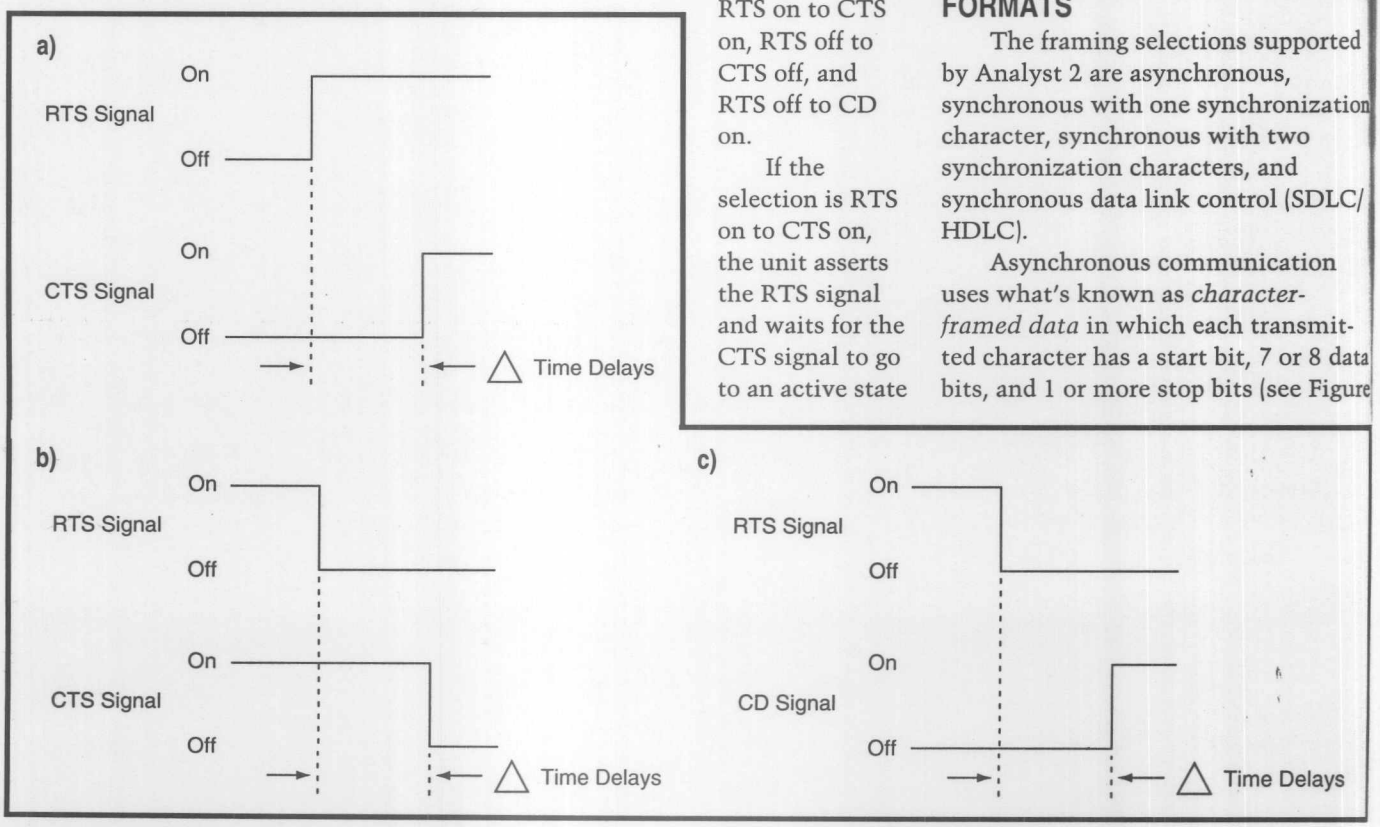


Figure 5—*The Analyst 2's time delay test is used to measure the internal delay on various control signal leads within the RS-232 interface. Options include a) RTS on to CTS on, b) RTS off to CTS off, and c) RTS off to CD on (c).*

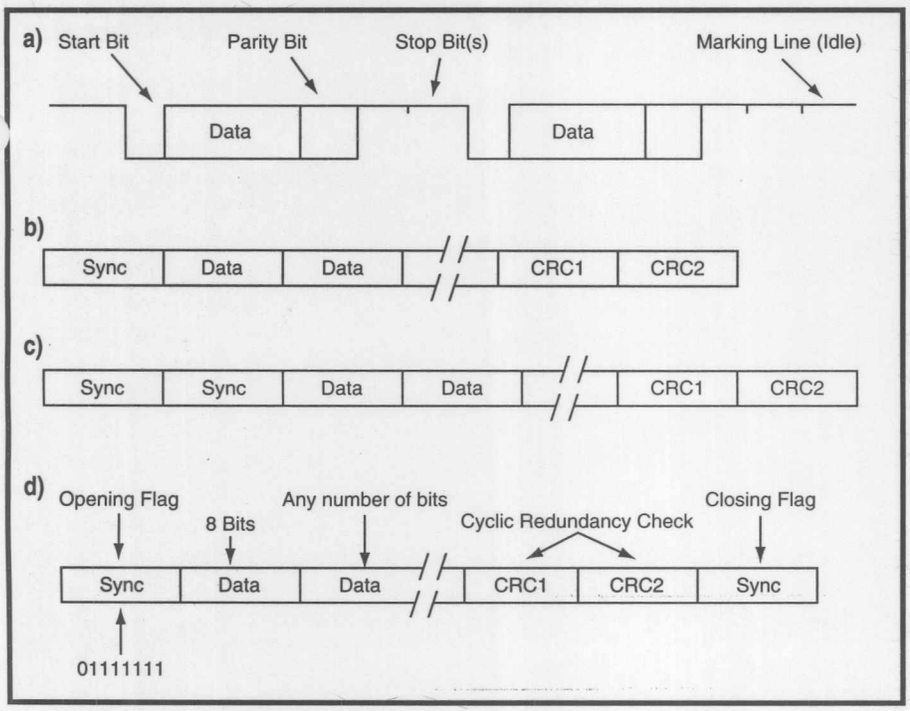


Figure 6—*Various serial communications methods include a) asynchronous, b) synchronous using one sync character, c) bisynchronous, and d) SDLC/HDLC/X.25 bit-oriented.*

unit can either use an internally generated clock or accept clocking from an external source. Additionally, it can recover the clock from an encoded data stream, such as nonreturn to zero Invert (NRZI), for reception and transmission if desired.

The available selections for clocking Analyst 2 are Internal, NRZI, and From DCE.

•Internal—this selection implies that no clocking signals will be present on the DB-25 interface. Instead, all clocking will be generated internally by the device and is used exclusively by asynchronous communications systems such as those on the COM1 and COM2 ports of a PC. After selecting Internal, the baud rate for the communications line has to be set to any of the standard values from 50 to 38,400 bps.

•NRZI—this selection implies that clocking is embedded within the data stream itself and is recovered using a digital phase-locked loop. This clocking information is then used to process the received bitstream into proper bit-timing periods. Most communications systems use non-return-to-zero (NRZ) encoding for bit transmission which preserves a one as a logic 1 and zero as a logic 0. In NRZI encoding, logic 1 is represented by no change in the signal polarity, and logic 0 is used to alter the polarity of the signal (see Figure 7). For Analyst 2 to process NRZI encoding properly, the communications speed for the line being tested must be entered. This value is used to set up the baud rate generators for a ×1 clock for all transmitted data and a ×32 clock for the input to the digital phase-locked loop for the receiver.

•From DCE—this selection implies that the communications line to which Analyst 2 is attached is providing clocking signals on the DB-25 interface leads (see Figure 8). These clocking signals are used both to receive and transmit data. In most cases, the transmit clocking is on interface pin 15, and the receive clocking is on interface pin 17. If this is not the case, Analyst 2 can be configured to derive clocking from either of the pins for both transmit and receive operation.

The setup for the data format must be entered after the clocking information has been entered. The menus are for Bits per Character, Parity (asynchronous only), Stop Bits

(asynchronous only), Bit Order, and Data Inversion.

•Bits per Character—this sets the number of bits used by the ZSCC to determine character boundaries. You can choose 8, 7, 6, or 5 bits.

•Parity—provides a means of error detection in asynchronous communications. It can be set to none, odd, or even.

•Stop Bits—this sets the number of stop bits used by the ZSCC to determine character boundaries in asynchronous systems. The number of Stop Bits can be set to 1, 1.5, or 2 bits.

•Bit Order—this sets the order in which the captured bits are received. Most transmission systems use a 1–8 (least significant bit first) transmission scheme. Analyst 2 can be set to either 1–8 or 8–1.

•Data Invert—this selection offers the option of decoding protected data by inverting each bit as it is captured. The setting is a toggle in which "yes" inverts the data and "no" captures it as it is. Few civilian communication systems invert data which is being transmitted.

If a synchronous data format has been selected, Analyst 2 prompts the user for the Sync Character, Drop Sync Character, and characters to be suppressed (if any).

•Sync Character—a specific bit pattern used by the ZSCC to establish character boundaries for the incoming data stream. The available selections are Sy_Sy, Dle_Sy, Flag, or User.

If the data format selected was BiSync, the default will be the Sy_Sy bit pattern, which loads the sync pattern detector of the ZSCC with a 16-bit value. If the code selection is ASCII, the value becomes a hexadecimal 1616, and if EBCDIC, the value is the hexadecimal 3232. Some Bisync systems use a Data Link Escape (DLE) character before the sync character for system-bit synchronization.

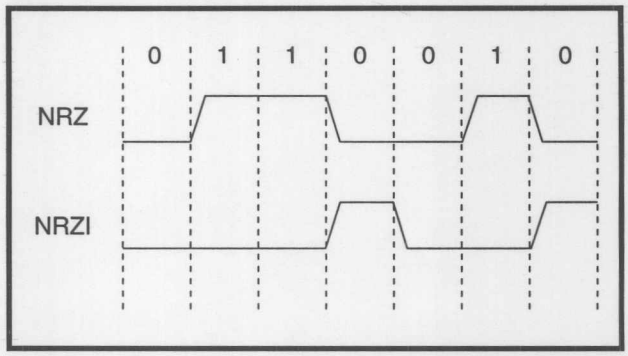


Figure 7—*In NRZ encoding, a 1 is represented by a high level and a 0 is represented by a low level. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level.*

To capture data properly in this type of system, the user needs to select the Dle_Sy menu item. If the code selection is ASCII, the 16-bit sync pattern detector of the ZSCC is loaded with a hexadecimal 1016, and if EBCDIC, the value is a hexadecimal 1032.

With SDLC/HDLC format, the default flag marks the beginning or end of the transmission frame. The sync pattern detector will be loaded with a hexadecimal 7E.

With the 1_Sync data format, Analyst 2 requires the entry of a two-digit hexadecimal number to be used as the sync character. This enables Analyst 2 to be used in proprietary communication systems which use nonstandard synchronization bit patterns.

•Drop Sync—set the bit pattern for the ZSCC so it can look for the next sync character in the data stream. If the selection is 1, the ZSCC will begin searching for a new sync character as soon as the line goes to a marking state. This state is commonly referred to as the *idle line* condition on a communications line.

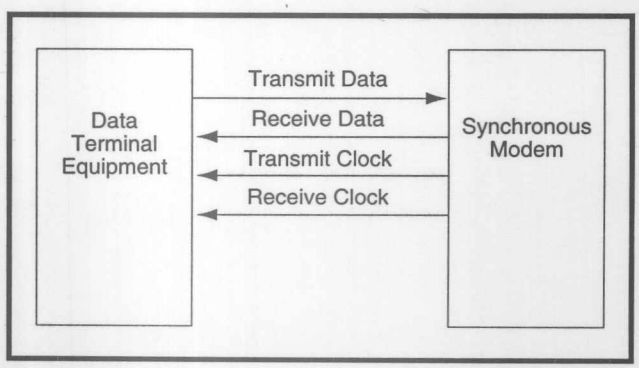


Figure 8—*In a synchronous setup, the modem provides the clocking information to the DTE (the clocks are both ×1).*

•User—this selection enables the user to enter a two-character hexadecimal value. When this value is detected in the incoming data stream, the ZSCC enters hunt mode and begins looking for the next sync character.

In some synchronous environments, a specific character is used to signify a marking line instead of a signal level. In this case, Analyst 2 needs to be configured to suppress that character from being transferred into the capture memory. The available selections are mark, space, none, sync character, and user defined.

The display update speed may seem relatively slow compared to the data speed. However, data capture can take place at 38,400 bps, a rate exceeding LCD readability. As the LCD updates, the capture RAM continues to fill according to the selected mode of operation. The available modes are Continuous, Buffer, Error Stop (asynchronous only), Signal, or User.

•Continuous—captures data until the capture RAM is filled. When this occurs, an alarm sounds once and Analyst 2 begins ignoring the data on the line. The LCD starts incrementing at a faster rate since it now has the priority. When the LCD gets to the end of the capture RAM, the alarm beeps three times and the unit begins capturing data from the communications line after a 0.5-s pause. The unit exits from the monitor mode if any key is pressed during the pause.

•Buffer—captures data until the capture RAM is filled. When this occurs, the unit beeps three times and begins ignoring data on the line. When the LCD gets to the end of the capture RAM, the unit waits for the Rset key to be pressed before beginning another capture.

•Error Stop (Er Stop)—prompts the user for more information about the type of error: Parity (P), Framing (F),

Parity and Framing (P,F), Break (B), Parity and Break detect (P,B), Framing and Break (F,B), and Parity, Framing and Break detect (P,F,B). The data line is continually monitored for an error condition occurrence after a selection has been made. Data capture stops on detection of the selected error.

•Signal—enables Analyst 2 to be configured to monitor a specific signal lead on the RS-232 interface for a transition. The unit is therefore able to function as a glitch catcher. The available selections are RTS (pin 4), CTS (pin 5), DSR (pin 6), and CD (pin 8). After selecting a signal to watch, the user must enable the trap function. Care should be exercised in enabling the trap function since this setting is stored in the nonvolatile configuration RAM and remains in effect at all times. The polarity of the signal to be watched must be entered after enabling the trap function. Either edge can be selected as the trigger.

•User—enables Analyst 2 to be programmed to either start or stop capture at the occurrence of a specific user-entered pattern. The unit prompts the user for up to 8 hexadecimal characters. "Don't cares" can be entered as "*".

CONCLUSION

Analyst 2 is a very powerful device for debugging a serial communications problem, but it also becomes a very handy tool around the shop for all sorts of things. For examples, RS-485 twisted-pair systems can be debugged by constructing a simple RS-232–to–RS-485 converter. Serial links between processors using a three-wire interface (such as that supported by the 8051 and other microcontrollers) can be debugged with a simple TTL–to–RS-232 converter.

Good luck with the kit and happy bug hunting!

Bill Payne holds a B.S. in Computing Sciences from the University of Oklahoma, College of Electrical Engineering. He has 12 years of experience in the design of computer-based equipment. He holds two semiconductor patents and has four others pending. He is also a Novell Certified Netware Engineer (CNE). He can be reached at (214) 487-7074.

SOURCE

A complete kit including all components and a four-layer circuit board is available from Payne Research. They can be reached at (214) 487-7074.

SOFTWARE

Software for this article is available from the Circuit Cellar BBS and on Software On Disk for this issue. Please see the end of "ConnecTime" in this issue for downloading and ordering information.

I R S

404 Very Useful
405 Moderately Useful
406 Not Useful